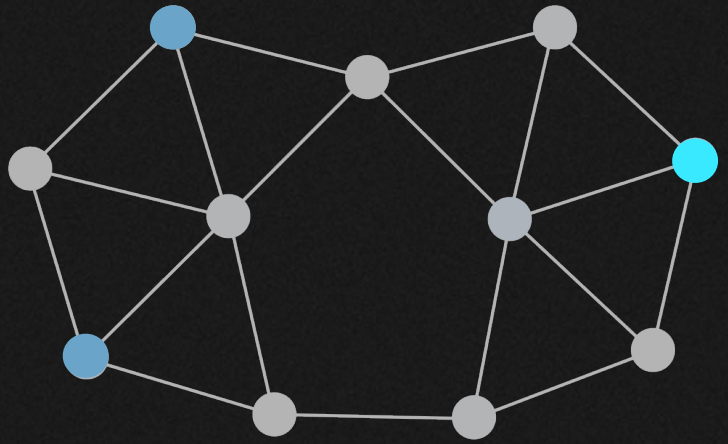
Distributed Systems

HELLO WORLD



Introduction to MPI

Message Passing Interface (MPI)

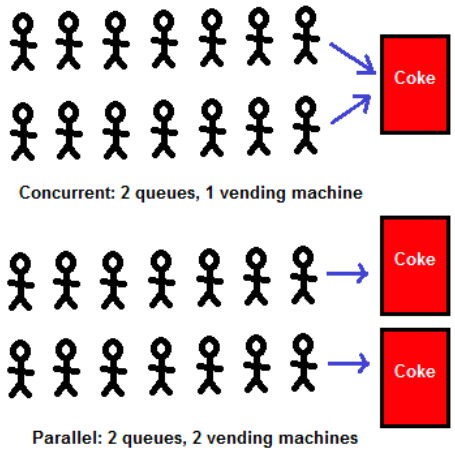
- Standard for Message Passing Computation: a collection of processes communicating with messages.
- Cross Platform (desktop to supercomputer)
- Language Agnostic (C, C++, FORTRAN…)

Message Passing Interface (MPI)

- MPI is a library, not a language.
- Specifies: the names, calling sequences and results of functions or subroutines to be called from C/C++ or Fortran programs, and the classes and methods that make up the MPI C++ library.
- Programs that users write are compiled with ordinary compilers and linked with the MPI library.

Parallelism

- Task parallelism: the work of a global problem can be divided into a number of independent tasks, which rarely need to synchronize.
- E.g. Monte Carlo simulations or numerical integration



Disadvantages/Issues

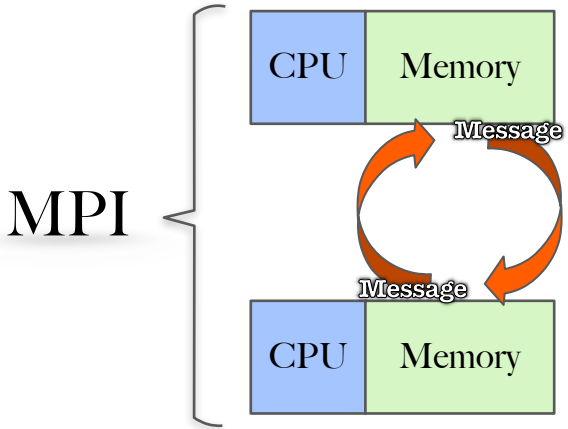
- No free lunch - can’t just “turn on” parallel
- Parallel programming requires work
  - Code modification - always
  - Algorithm modification - often
  - New sneaky bugs - you bet
- Speedup limited by many factors

Computer Architecture

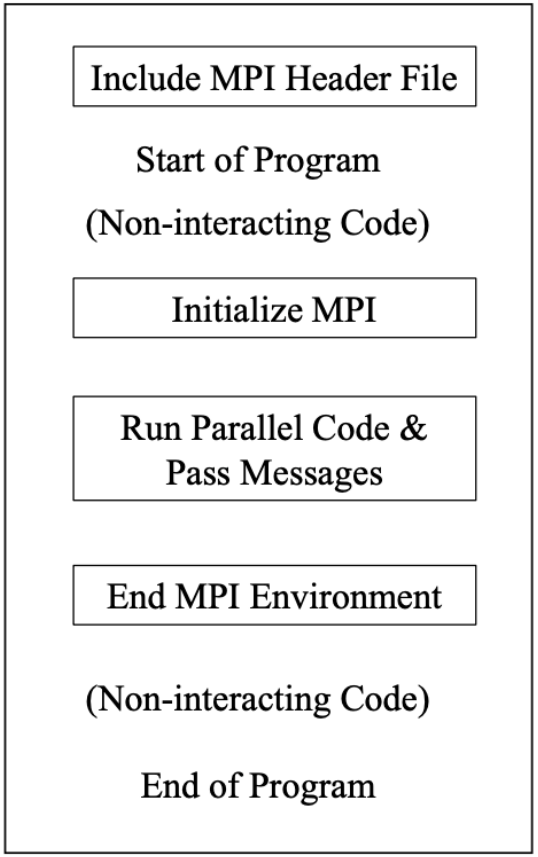
- As you consider parallel programming understanding the underlying architecture is important
- Performance is affected by hardware configuration
  - Memory or CPU architecture
  - Numbers of cores/processor
  - Network speed and architecture

MPI is a library

- MPI is a library specification for the message passing interface, proposed as a standard.
  - independent of hardware;
  - not a language or compiler specification;
  - not a specific implementation or product.
- A message passing standard for portability and ease-of-use.
- Designed for high performance.



MPI Program - Basics




```
#include <mpi.h>

int main(int argc, char *argv[])
{

MPI_Init(&argc, &argv);

.
.     // Run parallel code
.

MPI_Finalize();   //End MPI Environment

return 0;

}
```

MPI Common Terms

Communicator

- A group of MPI processes with a name (context).
- Processes identified by rank (only meaningful within particular communicator).
- Mechanism to identify subset of processes.

`MPI_COMM_WORLD`

- Predefined default communicator
- Contains all of the MPI Processes
- Size = N_{procs}

MPI: Most important functions

`MPI_Init`	Initiate an MPI computation
`MPI_Finalize`	terminate the MPI computation and clean up
`MPI_Comm_size`	how many processes participate in a given MPI communicator?
`MPI_Comm_rank`	which one am I? (Number between 0 and size-1)
`MPI_Send`	send a message to a particular process within an MPI communicator
`MPI_Recv`	receive a message from a particular process within an MPI communicator
`MPI_Bcast`	send a message to all processes within an MPI communicator
`MPI_Reduce`	collect overall data and reduce to a single datum using a specified operation
`MPI_Gather`	collect overall data from all processes within an MPI communicator
`MPI_Scatter`	divide a large amount of data among all processes within an MPI communicator

Output

- Output to screen is not ordered
- All processes are trying to write to screen simultaneously.
- The OS opts for an ordering.
- For organized output, starting from the first process, we can rewrite our program (E.g. Using MPI_Barrier)

Native install

Linux

Read Document on Moodle → MPI_setup - Linux.pdf

```
sudo apt-get install build-essential libopenmpi-dev openmpi-bin openmpi-doc
```

(or whatever package manager you use)

Mac

Read Document on Moodle → MPI_setup - Mac.pdf

http://brew.sh

In a terminal: `brew install open-mpi`

Caveats:

1. *"There are not enough slots available in the system to satisfy the x slots"*

   Add the following argument to your mpirun call: `--oversubscribe`

   Ex: `mpirun -n 4 --oversubscribe ./myfile.exe`

2. You might need to elevate (sudo) the *mpirun* command

Windows

Read Document on Moodle → MPI_Setup - Windows.pdf

- ☑ Install VisualStudio 2022 with C++ workspace
- ☑ Install ms-mpi
- ☑ Configure VS to use MSMPI library

Virtualbox

Only if already familiar with Linux !

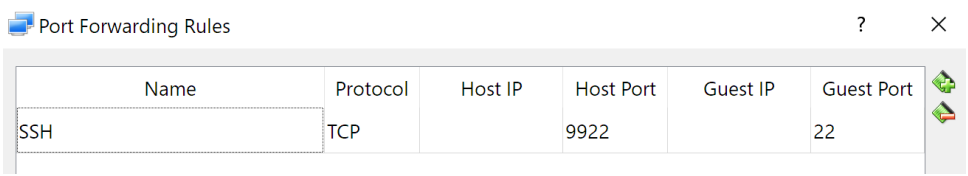
Create a VM

Download & install VirtualBox.

Install Ubuntu Server LTS in a VM.

Allow SSH access to VM.

Network->Advanced->Port Forwarding



Use putty to connect to VM.

127.0.0.1:9922

Install software:

```
sudo apt-get install build-essential libopenmpi-dev openmpi-bin openmpi-doc
```

Tweak Linux

Add the following lines to the following two files:

```
sudo nano /etc/openmpi/openmpi-mca-params.conf
    btl_base_warn_component_unused = 0


sudo nano /etc/openmpi/openmpi-default-hostfile
    localhost slots=4
```

First App

Important

Complete the instructions for installation and setup for your own OS on Moodle before attempting to run the following programs.

Hello World

Create a hello world file: “hello.cpp”.

Add some code. →

```cpp
//This is a comment
#include <iostream> //preprocessor - looks for file, and puts it here

//The main() function is a special function, it's the begining
int main(int argc, char **argv) {
    //print a message to the screen
    std::cout << "Hello world!" << std::endl;
    return 0; //return
}
```

Compile: `mpic++ ./hello.cpp -o ./hello`

-or-

Build using IDE

Run: `mpirun ./hello`

-or-

`mpiexec ./hello`

Note: *mpirun*, *mpiexec*, and *orterun* are all synonyms for each other. Using any of the names will produce the same behavior.

What is mpirun?

mpirun [options] **<program>** [<args>]

- A command which controls several aspects of program execution in Open MPI.
- Uses the Open Run-Time Environment (ORTE) to launch jobs.
- If you are running under distributed resource manager software, such as Sun Grid Engine or PBS, ORTE launches the resource manager for you.

More info: http://www.sciencedirect.com/science/article/pii/S0167739X07000507

https://docs.oracle.com/cd/E19356-01/820-3176-10/ExecutingPrograms.html#50413574_62903

Note: *mpirun*, *mpiexec*, and *orterun* are all synonyms for each other. Using any of the names will produce the same behavior.

What is mpiexec?

mpiexec [options] **<program>** [<args>]

- A command which controls several aspects of program execution in Open MPI.
- Uses the Open Run-Time Environment (ORTE) to launch jobs.
- If you are running under distributed resource manager software, such as Sun Grid Engine or PBS, ORTE launches the resource manager for you.

More info: http://www.sciencedirect.com/science/article/pii/S0167739X07000507

https://docs.oracle.com/cd/E19356-01/820-3176-10/ExecutingPrograms.html#50413574_62903

Note: *mpirun*, *mpiexec*, and *orterun* are all synonyms for each other. Using any of the names will produce the same behavior.

What is orterun?

orterun [options] **<program>** [<args>]

- A command which controls several aspects of program execution in Open MPI.
- Uses the Open Run-Time Environment (ORTE) to launch jobs.
- If you are running under distributed resource manager software, such as Sun Grid Engine or PBS, ORTE launches the resource manager for you.

More info: http://www.sciencedirect.com/science/article/pii/S0167739X07000507

https://docs.oracle.com/cd/E19356-01/820-3176-10/ExecutingPrograms.html#50413574_62903

Compilation and Execution

Depends on your OS!

To compile your file: `mpic++ NameofFile.cpp -o NameofFile`

This will create an executable file with that name, that you can then run using the mpirun command:

```
mpirun -np (number of nodes) --oversubscribe (name of file)
```

Compilation and Execution Windows + VS

To compile your file: `Build Solution (ctrl+shift+B)`

This will create an executable file with the same name as the original .cpp file,
that you can then run using the mpiexec command:

```
mpiexec -np (number of nodes) --oversubscribe (name of file)
```

Compilation and Execution OS + Xcode

To compile your file: `Build Solution (⌘+B)`

This will create an executable file with the same name as the original .cpp file,
that you can then run using the mpirun command:

```
mpirun -np (number of nodes) --oversubscribe (name of file)
```

Open MPI App

Now that you've run a hello world using c++, try importing the MPI library and running using more than one node.

```cpp
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
    MPI_Init(NULL, NULL);
    int rank;
    int world;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &world);
    printf("Hello: rank %d, world: %d\n",rank, world);
    MPI_Finalize();
}
```

Create a new file: "hello2.cpp".

Compile: `mpic++ ./hello2.cpp -o ./hello2`

Run: `mpirun -np 4 --oversubscribe ./hello2`

Trouble?

Send() fails?

tcp_peer_send_blocking: send() to socket 12 failed

```
netstat -rn
route print
```

Should be the same IP as the first one in:

```
mpirun -report-uri - -n 4 ./a.exe
```

So disable NICs to make sure MPI uses your correct IP.

OpenCL errors when running in Xcode

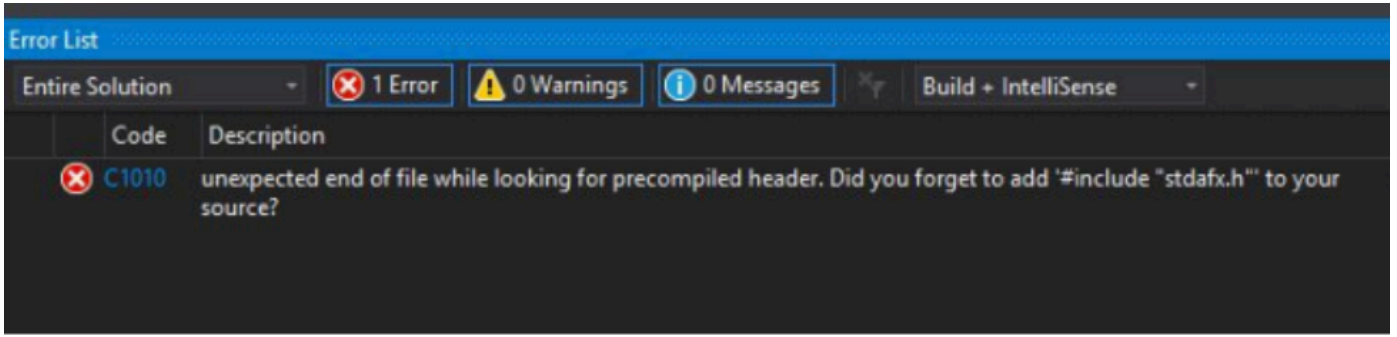
- You may ignore these, they only appear when running directly via the IDE.

```
2022-09-05 16:56:12.363081+0100 orterun[636:6500280] [CL_INVALID_OPERATION] : OpenCL Error :
    Failed to retrieve device information! Invalid enumerated value!

2022-09-05 16:56:12.364089+0100 orterun[636:6500280] [CL_INVALID_OPERATION] : OpenCL Error :
    Failed to retrieve device information! Invalid enumerated value!

2022-09-05 16:56:12.364143+0100 orterun[636:6500280] [CL_INVALID_OPERATION] : OpenCL Error :
    Failed to retrieve device information! Invalid enumerated value!
```

Linking Error Messages (Windows)

- If you get any linking error messages, it is most likely you're building for 64 bits yet specifying 32 bits linking libraries.
- Ensure you are using the correct linking libraries for your architecture
- You can check your architecture by going to Windows > Settings > System > About > System type, this should tell you your processor type.

Unexpected end of file C1010



- Go to Property Pages – C/C++ Precompiled Header then click Precompiled Header and select Not Using Precompiled Headers.
- Note: Precompiled headers can be turned off when you create a new project!

Add exceptions for...

Firewall

AV

LabWork - Week 1

Lab Work - Week 1

- Complete installation and run your first two Hello World programs.
  - ➡ One using only C++
  - ➡ One using C++ and MPI
- Start MPI_Notes tutorials